Linux ユーザーのための ネットワークセキュリティ

桃木　悟 (工学部機械システム工学科)
e-mail: momoki@net.nagasaki-u.ac.jp
古賀掲維 (工学部構造工学科)
e-mail: aoi@st.nagasaki-u.ac.jp

1　はじめに

MS-DOS しか使った事の無かった私たちが、「タコは財産である、育てようではないか」のスローガンに乗せられて Linux を使いはじめたのは約 5 年前のことです。当時、それほど注目されていなかった Linux ですが、最近では静かなブームとなっているようで、学内にも、既に使っている方や初めてみようと思っている方は少くないと思います。そこで、長崎の Linux ユーザー間の情報交換を行う事を目的として、本学情報処理センターのメーリングサービスに「長崎 Linux ユーザーズグループのメーリングリスト[1]」の開設を申請しました (ML への参加方法については、付録 A に簡単に説明します)。この ML を足掛りにして、長崎 Linux ユーザーズグループの設立までこぎつけたいと目論んでいますので、Linux を使っている、あるいは使いたいという方は是非参加していただくようよろしくお願いします。

今日のインターネットの状況は、私が Linux を始めた当時とくらべると、格段にその存在価値が向上する一方で、治安が悪化の一途をたどっています。Linux 等の UNIX 系 OS の場合、ネットワーク関連サービスが充実している分、侵入されやすく一旦侵入された場合に他人に迷惑を及ぼす可能性も高いため、最低限のセキュリティの設定は利用者の義務ともいえます。しかしながら、Linux を使っている (あるいは使おうと思っている) 方の多くは、コンピュータにそれ程詳しくないし極めるつもりもない、できるならネットワークの管理などはやりたくない[2]と思っているのではないしょうか。このレポートは、そんな Linux ユーザーがインターネットに接続する際に最低限気をつけるべき事について、注意を喚起する目的も兼ねて、私が知っている範囲で述べたものです。

2　セキュリティを意識したネットワークの設定

セキュリティの見地から最も重要な事は、何よりも新しいディストリビューションを使用する事です。最近のディストリビューションでは特に指定しない限り、tcp-wrapper 等の最低限のセキュリティ関連ツールがインストールされますので、古いバージョンのディストリビューションを使っているならば、最新のバージョンにアップして下さい。セキュリティ以外の面でも便利な点が多々ありますから損はないはずです[3]。

この章では、Linux コンピュータをインターネットに接続する際に、セキュリティの観点から、注意すべき事について説明します。ここで、対象とするコンピュータは、1) ネットサーフィン (例えば Netscape でホームページを見てまわることです、2) メールの読み書きですが、オプションとして 3) ssh による他のマシンからのリモートアクセス, および 4) プリンタあるいはファイルを他のマシンから使用できるようにする、以上のネットワークサービスが使用出来ることを想定します。ssh が使えることで X によるグラフィカルなアプリケーションのネットワークでの使用や scp によるファイルの転送サービスができますし、scp の

[1] nlug@ml.nagasaki-u.ac.jp
[2] 少くとも私はそうです。
[3] バージョンアップが難しいディストリビューションを使っているならこれを機に RedHat 系か Debian 系のものに変更しましょう。

ポート転送機能を用いて ftp サービス等も可能となります [1] から、想定した状況は今日使用される Linux マシンとしてはごく標準的なものだと思います。

ネットワークセキュリティについて悩みたくなければ、あたりまえの事ですが、ネットワーク経由のアクセスを基本的に禁止することです。そのための設定は、/etc/hosts.deny ファイルで行います。まず、ルートになって 好みのエディタで /etc/hosts.deny を開いて内容を確認しましょう。そこに ALL : ALL と書いた行がなければ追加します。

```
---------------------------------------------------------------------------
#
# hosts.deny    This file describes the names of the hosts which are
#               *not* allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
#
ALL : ALL
# End of hosts.deny.
---------------------------------------------------------------------------
```

例 1: secure な /etc/hosts.deny の設定例

最近の Linux ディストリビューションを使っているならば、例 1 の設定でネットワーク経由のアクセスをブロックできます[4]。ところが、このままでは、自分自身からのアクセスも禁止されていますから、日本語変換プログラムと辞書サーバーとの連携が正常に動作しない等の問題が発生します。UNIX のネットワーク対応プログラムの多くは、自分自身に対してもネットワーク経由でアクセスするからです。この問題を解決するために /etc/hosts.allow に ALL : 127.0.0.1 を追加します。127.0.0.1 というのは自分自身を意味する特殊なアドレスです。

```
---------------------------------------------------------------------------
#
# hosts.allow   This file describes the names of the hosts which are
#               allowed to use the local INET services, as decided by
#               the '/usr/sbin/tcpd' server.
#
ALL : 127.0.0.1
# End of hosts.allow.
---------------------------------------------------------------------------
```

例 2: /etc/hosts.allow の設定例

次に所望のネットワークサービスに必要な設定を加えます。1) のネットサーフィンについては何も設定する必要はありません。アクセスするのは自分です:-)。Netscape や Lynx をインストールして使うだけです。しかし、2) のメールの読み書きについては、設定が必要になるかもしれません。Windows 95 と同様にメー

[4] もちん，特殊なデーモンを起動していたり特殊な設定をしている場合はその限りではありません。

ルを他の POP/IMAP サーバ (例えば 教職員用に情報処理センターが提供しているメールサーバ[5]) から読みこむ場合には何も特別な設定は必要ありませんが、sendmail や smail でメールを直接受けとっている場合には、/etc/hosts.allow に

```
sendmail : ALL
```

または,

```
smail : ALL
```

を加えて下さい。これが無いと自分宛のメールが届きません。qmail の場合には、普通、tcpd 経由で起動しませんから /etc/hosts.deny が効いていないようなので何も指定しなくてもメールを受けとれます。

自分が管理しているコンピュータが他にもあるなら、そのマシンからもアクセスできてこそ UNIX というものです。そのために、何はともあれ ssh をインストールしましょう[6]。ディストリビューションによって異なりますが、バイナリパッケージが提供されているはずです[7]。例えば, Debian/GNU Linux の場合 non-US のディレクトリ [2] にバージョン 1.2.26 のパッケージが提供されています。ssh のインストールが終了したら, 他のコンピュータから ssh によるアクセスを許可するために、/etc/hosts.allow に

```
sshd : ALL
```

を加えて下さい。ここでは、出張先の不特定のコンピュータからアクセスすることを想定して ALL にしていますが、アクセス元のアドレスが限られている場合には、ALL でなくそのアドレスを書いた方がいいでしょう。例えば、133.45.xxx.2 と 133.45.xxx.3 にだけアクセスを許可する場合には、sshd : ALL の代わりに,

```
sshd : 133.45.xxx.2
sshd : 133.45.xxx.3
```

の 2 行を追加します。192.168.1. 以下の全てのコンピュータのアクセスを許可する場合には,

```
sshd : 192.168.1.
```

と書くこともできます。

2 台以上のコンピュータを使っているならば、ホームディレクトリのファイルを、nfs や samba で 共有すると便利です。またプリンタも全てのコンピュータから印刷ができるようにしたいものです。この場合も, 個々のサービスについて個別に許可するように /etc/hosts.allow を修正すれば良いのですが、nfs を許可するようなコンピュータに対しては全てのアクセスを許可する設定でも構わないと思います。例えば、コンピュータ 133.45.xxx.3 と コンピュータ 133.45.xxx.4 に対してファイルやプリンタを共有するならば /etc/hosts.allow に,

```
ALL : 133.45.xxx.4
ALL : 133.45.xxx.3
```

[5] net.nagasaki-u.ac.jp
[6] telnet や rlogin の事は忘れましょう。
[7] 提供されていないならばディストリビューションの変更を考えてもいいかもしれません。それぐらい大事なツールです.

を追加して下さい。もちろん 133.45.xxx.4 と 133.45.xxx.3 は自分の責任でしっかりと管理して下さい。

最後に、以上の設定を有効にするためにコンピュータを再起動させて下さい。Linux の名誉のために付け加えますが、必ずしも再起動が必要なわけではなく、設定を有効にするための方法はいくつか存在します。しかし、これらの設定の変更は頻繁に行うものではありませんから、設定を有効にする呪文を覚えるよりも `shutdown -r now` や Ctrl+Alt+Del で再起動させるのが、確実で Linux ユーザーらしい方法といえるでしょう。

3 IP マスカレードによるローカルネットワーク

私の場合、タコに毛がはえた程度の能力しかないにもかかわらず、数年前から専用ルータを使って研究室専用のサブネット (133.45.xxx.) を管理しています。所属する研究室において、他が Windows 95/98 や Macintosh ユーザーばかりだと、Linux ユーザーレベル (!) の能力しかない場合でも私のように研究室内のコンピュータの管理[8]がまわってくる可能性があります。その場合、ネットワーク内のコンピュータのセキュリティも面倒をみることになるでしょう。一般に、セキュリティに関する設定をきつくすると、それまで使えていたサービスが使えなくなる事が多々あります。大抵は、ちょっとした設定ミスなのですが、自分が使っていないコンピュータの場合には、原因解明は予想以上に困難です。かといって、今日のインターネット事情ではセキュリティに手を抜くことは許されませんから、私のような B 級の管理者にとっては頭がいたい問題です。

幸いな事に、近年の Linux におけるファイアーウォール関連のサポートは充実していまして、IP マスカレード程度でしたら比較的簡単に実現できます。ですから、先の研究室ネットワークセキュリティの問題は、Linux で IP マスカレード機能を使った即席ファイアーウォールを作成して、ローカルネットワークを構築するのが、トータルでは最も楽な解決法ではないでしょうか。最近、バージョン 2.2 のカーネルがリリースされたこともありますので、心機一点して、実験的にですが ..., IP マスカレード機能を使った即席ファイアーウォールの作成、およびパケットの転送によるファイアーウォール内のコンンピュータでのメールの受信と ssh による外からのアクセスの実現が確認できましたので、ここにその概要を簡単に報告します。どうしても私が使用している Debian/GNU システムに偏った記事になってしまいますが、あらかじめ御了承下さい。なお、このレポートが手元に届くころには、現在管理しているサブネットを廃止して IP マスカレードを使ったローカルネットワークを作成している予定です。

IP マスカレードを使うには、まず 2 枚のネットワークカードを正しく設定しないといけません。第一のハードルはそれぞれのカードをカーネルに認識させる事ですが、それぞれ種類の異なるカードを用いれば、1 枚の場合と同じ作業 (通常はモジュールの設定) をそれぞれのカードに対して行うだけで済ませることができます。次に、ソフトウェア的な設定ですが、最低でも 2 枚目の分は手で設定ファイルに追加する必要があります。Debian/GNU Linux の場合は、`/etc/init.d/network` を例 3 のように変更する必要があります。

```
-------------------------------------------------------------
#! /bin/sh
ifconfig lo 127.0.0.1
route add -net 127.0.0.0

#
# eth0 : インターネット側
```

[8] ただの雑用です

```
#
DEV=eth0
IPADDR=133.45.xxx.yy
NETMASK=255.255.255.0
NETWORK=133.45.xxx.0
BROADCAST=133.45.xxx.255
GATEWAY=133.45.xxx.1
ifconfig ${DEV} ${IPADDR} netmask ${NETMASK} broadcast ${BROADCAST}
route add -net ${NETWORK}
[ "${GATEWAY}" ] && route add default gw ${GATEWAY} metric 1

#------------------------------------------------------------
#
# eth1 : ローカルネット側
#
DEV=eth1
IPADDR=192.168.1.1
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
ifconfig ${DEV} ${IPADDR} netmask ${NETMASK} broadcast ${BROADCAST}
route add -net ${NETWORK}
```

例 3 : Network の設定例 (`/etc/init.d/network` on Debian/GNU Linux)

次にファイアーウォールの作成に取りかかるわけですが、せっかくですからカーネル 2.2.x をコンパイルしましょう[9]。カーネル 2.0.x でも IP マスカレードは可能ですが、新しいカーネルではファイアーウォールに関する設定が大幅に変更されていますので、ここでは新しいカーネルについてしか述べません。カーネルのコンパイル方法そのものは従来と変わりありません。ただ設定項目が大幅に増えているだけです。カーネルのコンパイルについては、多くの参考文献 [3, 4] がありますので参考にして下さい。ファイアーウォールを作るわけですから、CONFIG_FIREWALL、CONFIG_IP_FIREWALL, CONFIG_IP_MASQUERADE といった項目は必ず y に設定して下さい [5]。

設定には ipchains と ipmasqadm を使います [5]。いずれも Network と IP-masquerade に関するパッケージを最新のものを導入すればいずれも含まれているものです。(ipmasqadm については `http://juanjox.home.ml.org` からバイナリを入手する必要があるかもしれません [5]。) Debian/GNU Linux の場合、いずれも netbase パッケージに含まれています[10]。それだけではなくて、ipmasq パッケージをインストールすると 1) ローカルネットワーク側 (`192.168.1.1`) からインターネット側 (`133.45.xxx.yy`) へは、全てのアクセスについてマスカレードする, 2) インターネット側からローカルネットワーク側へは全てのアクセスを拒否する、の設定をしてくれます[11]。この 1), 2) の設定を手動で行うには、それぞれ、

```
ipchains -A forward -j MASQ -s 192.168.1.0/24 -d 0.0.0.0/0
```

[9] このレポートが手元に届くころにはバイナリも入手できるようになっているかもしれません
[10] 最新の potato の場合
[11] うーん、便利。

```
ipchains -P forward DENY
```

を実行する必要があるようです。

次に、ファイアーウォールの外から内側のコンピュータ (例えば 192.168.1.32) へ ssh アクセスを可能にするための設定について説明します (あたりまえの事ですが、192.168.1.32 のセキュリティはしっかりと設定して下さい)。具体的には、起動スクリプトのどこかに、以下のコマンドを記述します。Debian/GNU Linux では /etc/ipmasq.conf に 記述するのが最もスマートな方法です。

```
#!/bin/sh
FWMARK="1"
PORT=22  # 22 -> ssh
FIREWALL_ADR="133.45.xxx.yy"
RECIEVER_ADR="192.168.1.32"
ipchains  -I input -p tcp -y  -d $FIREWALL_ADR/32 $PORT -m $FWMARK
ipmasqadm mfw  -A  -m $FWMARK -r $RECIEVER_ADR   $PORT
```

ファイアーウォールの内側のコンピュータ (例えば, 192.168.1.33) でメールを受けとるための設定も ssh の場合とほぼ同様で、以下のコマンドを先のスクリプトに追加すれば OK です。

```
FWMARK="2"
PORT=25  # 25 -> smtp
FIREWALL_ADR="133.45.xxx.yy"
RECIEVER_ADR="192.168.1.33"
ipchains  -I input -p tcp -y  -d $FIREWALL_ADR/32 $PORT -m $FWMARK
ipmasqadm mfw  -A  -m $FWMARK -r $RECIEVER_ADR   $PORT
```

ssh の場合との違いは、PORT の値を 25 に, FWMARK の値を一つ増やして 2 に変更の 2 点です。以上の設定の結果、133.45.xxx.yy 宛てのメールはパケットレベル (と言うのかな) で 192.168.1.33 に転送されますから、ファイアーウォールに余分なユーザーを作る必要はありません。駆け足での説明になりましたが、以上で全ての作業は終わりです。再起動させれば、所望の即席ファイアーウォールの完成です。

私自身ネットワークについては初心者の域を越えていませんので、説明不足な点やおかしい点が多々あるかと思います。わからない点、疑問な点がありましたら遠慮せずに長崎 Linux ユーザーズグループ ML の方へメールを出して下さい。できる限り回答したいと思っています。最後に、くどいようですが、ここで説明したことは最低限の設定であることをくれぐれも忘れないで下さい。それではセキュリティに気をつけて、お互い快適な Linux life を送りましょう。次は、是非、nlug@ml.nagasaki-u.ac.jp でお会いしたいですね。

参考文献

[1] 島　慶一, Secure Shell (II), UNIX マガジン 98 年 7 月号

[2] www.debian.or.jp 等 (学内: ftp://lostwind.st.nagasaki-u.ac.jp/pub/linux/debian)

[3] 前田輝雄, Linux 2.1 カーネルガイド, フキ出版, 1998.

[4] http://www.linux.or.jp/kernel.html

[5] Paul Russell, Linux IPCHAINS-HOWTO
(http://metalab.unc.edu/LDP/HOWTO/IPCHAINS-HOWTO.html)

A 長崎 Linux ユーザーズグループのメーリングリスト

ここでは、長崎 Linux ユーザーズグループのメーリングリストへの参加申請法を簡単に説明します。まず,

majordomo@ml.nagasaki-u.ac.jp

宛てに,

subscribe nlug

と書いたメールを送ります. すると, 確認のメールが送られてきますので, その中にある.

auth xxxxxxxx subscribe nlug 参加希望者のメールアドレス

という行を同じアドレスに送り返します (xxxxxxxx は 16 進数です). これで, ML への登録は完了です。
次に ML へメールを出すには, nlug@ml.nagasaki-u.ac.jp 宛てにメールを出します. その他, 基本的な操作は, majordomo のマニュアル等を参考にされて下さい.